

2.1 Budowa i zawartość programu NC



Wytyczną dotyczącą budowy programu obróbki jest DIN 66025.

Program NC / program obróbki składa się z szeregu **bloków NC** (patrz poniższa tablica). Każdy blok przedstawia jeden krok obróbczy. W bloku są pisane instrukcje w formie **słów**. Ostatni blok w ciągu obróbkowym zawiera specjalne słowo oznaczające **koniec programu**: **M2, M17** wzgl. **M30**.

Blok	Słowo	Słowo	Słowo	...	;Komentarz
Blok	N10	G0	X20	...	;1. blok
Blok	N20	G2	Z37	...	;2. blok
Blok	N30	G91	;...
Blok	N40	
Blok	N50	M30	;koniec programu (ostatni blok)



Nazwy programów

Każdy program ma własną nazwę, która przy sporządzaniu programu może zostać dowolnie wybrana przy przestrzeganiu następujących warunków (oprócz formatu taśmy dziurkowanej):

- pierwsze dwa znaki muszą być literami (również podkreślnik jest literą)
- ponadto litery i cyfry

Przykład: _MPF100 albo
 WAŁEK albo
 WAŁEK_2

Tylko pierwsze **24** znaki identyfikatora programu są wyświetlane na NC.

Format taśmy dziurkowanej

Nazwy plików:

1. Nazwy plików mogą zawierać znaki 0 . . . 9, A . . . Z, a . . . z albo _ i mogą mieć długość maksymalnie 24 znaki.
2. Nazwy plików muszą posiadać rozszerzenie o długości 3 znaków (_xxx).
3. Dane w formacie taśmy dziurkowanej mogą być sporządzane na zewnątrz i opracowywane przy pomocy edytora. Nazwa pliku, która jest wewnętrznie zapisana w pamięci NC, rozpoczyna się od "_N_". Plik w formacie taśmy dziurkowanej rozpoczyna się od %<nazwa>, "%" musi znajdować się w pierwszej kolumnie pierwszego wiersza.

Przykłady:

%_N_WAŁEK123_MPF = program obróbki WAŁEK123

albo

%kołnierz3_MPF = program obróbki kołnierz3



Dalsze informacje o przesyłaniu, sporządzaniu i zapisywaniu programów obróbki znajdziecie w:

/BA/, Instrukcja obsługi, punkt "Zakres czynności obsługowych program" i "Zakres czynności obsługowych usług"

2.2 Elementy językowe języka programowania

Zasób znaków

W celu sporządzania programów NC są do dyspozycji następujące znaki:

Duże litery

A, B, C, D, E, F, G, H, I, J, K, L, M, N,
(O), P, Q, R, S, T, U, V, W, X, Y, Z

Należy przy tym przestrzegać:

litery "O" nie mylić z cyfrą "0".

Małe litery

a, b, c, d, e, f, g, h, i, j, k, l, m, n,
o, p, q, r, s, t, u, v, w, x, y, z

Cyfry

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

2.6 Przykład programowania, toczenie



Małe i duże litery nie są rozróżniane.

Znaki specjalne

%	Znak początku programu (tylko do sporządzania programu na zewnętrznym PC)
(Branie w nawiasy parametrów albo w wyrażeniach
)	Branie w nawiasy parametrów albo w wyrażeniach
[Branie w nawiasy adresów albo indeks tablicy
]	Branie w nawiasy adresów albo indeks tablicy
<	Mniejsze niż
>	Większe niż
:	Blok główny, zakończenie etykiety, operator powiązania
=	Przyporządkowanie, część równania
/	Dzielenie, maskowanie bloku
*	Mnożenie
+	Dodawanie
-	Odejmowanie, znak ujemny
"	Cudzysłów, identyfikator łańcucha znaków
^	Przecinek górny, identyfikator specjalnych wartości liczbowych: heksadecymalnych, binarnych
\$	Własne oznaczenie zmiennych systemu
_	Podkreślnik, zaliczany do liter
?	zarezerwowano
!	zarezerwowano
.	Kropka dziesiętna
,	Przecinek, znak rozdzielający parametry
;	Początek komentarza
&	Znak formatowania, takie samo działanie jak spacja
L _F	Koniec bloku
Tabulator	Znak rozdzielający
Spacja	Znak rozdzielający (pusty)



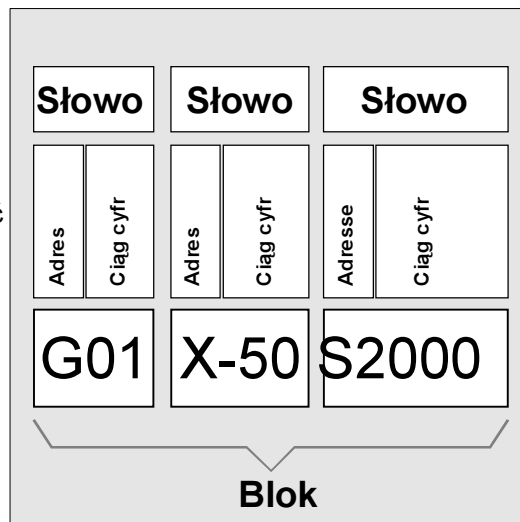
Nie dające się przedstawić znaki specjalne są traktowane jak spacje.

Słowa

Programy NC składają się z bloków, bloki składają się ze słów.

Słowo "języka NC" składa się z jednego znaku adresowego i cyfry wzgl. ciągu cyfr, który przedstawia wartość arytmetyczną.

Znakiem adresowym słowa jest zazwyczaj litera. Ciąg znaków może zawierać znak liczby i kropkę dziesiętną, przy czym znak liczby znajduje się zawsze między literą adresową i ciągiem cyfr. Znak dodatni (+) nie musi być pisany.



Bloki i ich budowa

Program NC składa się z poszczególnych bloków, blok składa się z reguły z (wielu) słów.

Blok powinien zawierać wszystkie dane do wykonania kroku roboczego a kończy się znakiem "LF" (LINE FEED = nowy wiersz).



Znak "LF" nie musi być pisany; jest on automatycznie wytwarzany przez przełączenie wiersza.

Długość bloku

Blok może

- do wersji oprogramowania 3.x zawierać maksymalnie **242** znaki
- **od wersji oprogramowania 4** maksymalnie **512** znaków (łącznie z komentarzem i znakiem końca bloku "LF")

2.6 Przykład programowania, toczenie



Zazwyczaj na aktualnym wyświetleniu bloków na ekranie są pokazywane maksymalnie trzy bloki po maksymalnie 66 znaki. Komentarze są również wyświetlane. Komunikaty są wyświetlane we własnym oknie komunikatów.

Kolejność słów w bloku

Aby budowa bloku była przejrzysta, słowa powinny być uporządkowane następująco:

Przykład:

N10 G... X... Y... Z... F... S... T... D... M... H...

<i>Adres</i>	<i>Znaczenie</i>
N	Adres numeru bloku
10	Numer bloku
G	Warunek drogowy
X, Y, Z	Informacja o drodze
F	Posuw
S	Prędkość obrotowa
T	Narzędzie
D	Numer korekcji narzędzia
M	Funkcja dodatkowa
H	Funkcja pomocnicza



Niektóre adresy mogą w ramach jednego bloku być stosowane również wielokrotnie (np.: G..., M..., H...).

Blok główny/blok pomocniczy

Rozróżnia się dwa rodzaje bloków:

- **bloki główne** i
- **bloki pomocnicze**

W bloku głównym muszą być podane wszystkie słowa, które są wymagane, aby móc uruchomić przebieg pracy od odcinka programu rozpoczynającego się od bloku głównego.



Bloki główne mogą znajdować się zarówno w programach głównych jak i w podprogramach. Sterowanie nie sprawdza, czy blok główny zawiera potrzebne informacje.

Numer bloku

Bloki główne są oznakowywane numerem bloku głównego. Numer bloku głównego składa się ze znaku ":" i dodatniej liczby całkowitej (numer bloku). Numer bloku znajduje się zawsze na jego początku.



Numery bloków głównych muszą być jednoznaczne w ramach programu, aby przy poszukiwaniu uzyskać jednoznaczny wynik.

Przykład: :10 D2 F200 S900 M3

Bloki uboczne są oznakowywane przez numer bloku ubocznego. Numer taki składa się ze znaku "N" i dodatniej liczby całkowitej (numer bloku). Numer bloku znajduje się zawsze na jego początku.

Przykład: N20 G1 X14 Y35
 N30 X20 Y40



Numery bloków pomocniczych muszą być jednoznaczne w ramach programu, aby przy poszukiwaniu uzyskać jednoznaczny wynik.



Kolejność numerów bloków jest dowolna, zalecana jest jednak numeracja rosnąca.

Można programować bloki NC również bez numerów.

2.6 Przykład programowania, toczenie

Adresy

Adresy są stałym albo nastawnym określeniem osi (X,Y, ...), prędkości obrotowej wrzeciona (S), posuwu (F), promienia okręgu (CR) itd.

Przykład:

N10 X100

Ważne adresy

<i>Adres</i>	<i>Znaczenie (nastawienie standardowe)</i>	<i>Uwagi</i>
A=DC (...)	Oś obrotowa	nastawny
A=ACP (...)		
A=ACN (...)		
ADIS	Odstęp ścięcia dla funkcji ruchu po konturze	stały
B=DC (...)	Oś obrotowa	nastawny
B=ACP (...)		
B=ACN (...)		
C=DC (...)	Oś obrotowa	nastawny
C=ACP (...)		
C=ACN (...)		
CHR=...	Sfazowanie narożnika konturu	stały
D...	Numer ostrza	stały
F...	Posuw	stały
FA[oś]=... wzgl.	Posuw osiowy	stały
FA[wrzeciono]=... wzgl.		
[SPI(wrzeciono)]=...		
G...	Warunek drogowy	stały
H...	Funkcja pomocnicza	stały
H=QU (...)	Funkcja pomocnicza bez zatrzymywania odczytu	
I...	Parametry interpolacji	nastawny
J...	Parametry interpolacji	nastawny
K...	Parametry interpolacji	nastawny
L...	Wywołanie podprogramu	stały
M...	Funkcja dodatkowa	stały
M=QU (...)	Funkcja dodatkowa bez zatrzymywania odczytu	
N...	Blok uboczny	stały
OVR=...	Ręczna zmiana ruchu po konturze	stały
P...	Liczba przebiegów programu	stały
<i>Adres</i>	<i>Znaczenie (nastawienie standardowe)</i>	<i>Uwagi</i>
POS[oś]=...	Oś pozycjonowania	stały
POSA[oś]=...	Oś pozycjonowania poza granicę bloku	stały

SPOS=...	Pozycja wrzeciona	stały
SPOS[n]=...		
SPOSA=...	Pozycja wrzeciona poza granicę bloku	stały
SPOSA[n]=...		
Q...	Oś	nastawny
R0=... do Rn=...	• Parametr obliczeniowy, n jest nastawne poprzez daną maszynową (standard 0 - 99)	stały
R...	• Oś (od wersji opr. 5.1)	nastawny
RND	Zaokrąglenie narożnika konturu	stały
RNDM	Zaokrąglenie narożnika konturu (możliwie)	stały
S...	Prędkość obrotowa wrzeciona	stały
T...	Numer narzędzia	stały
U...	Oś	nastawny
V...	Oś	nastawny
W...	Oś	nastawny
X...	Oś	nastawny
X=AC(...)	" absolutnie	
X=IC(...)	" przyrostowo	
Y...	Oś	nastawny
Y=AC(...)		
Y=IC(...)		
Z...	Oś	nastawny
Z=AC(...)		
Z=IC(...)		
AR+=...	Kąt rozwarcia	nastawny
AP=...	Kąt biegunowy	nastawny
CR=...	Promień okręgu	nastawny
RP=...	Promień biegunowy	nastawny
:...	Blok główny	stały



"stały"

Te identyfikatory adresowe są do dyspozycji dla określonej funkcji.



Producent maszyny (MH2.1)

"nastawny"

Tym adresom producent maszyny może poprzez daną maszynową przyporządkować inną nazwę.

2.6 Przykład programowania, toczenie

Adresy działające modalnie / pojedynczymi blokami

Adresy działające modalnie zachowują z zaprogramowaną wartością tak długo swoją ważność (we wszystkich następnych blokach), aż pod tym adresem zostanie zaprogramowana nowa wartość.

Adresy działające pojedynczymi blokami obowiązują tylko w tym bloku, w którym są zaprogramowane.

Przykład:

N10 G01 F500 X10

N20 X10 ;posuw działa tak długo, aż
zostanie wprowadzony nowy.

Adresy z rozszerzeniem osiowym

W przypadku adresów z osiowym rozszerzeniem nazwa osi jest w nawiasach kwadratowych po adresie, który ustala przyporządkowanie do osi.

Przykład: FA[U]=400;
posuw specyficzny dla osi U

Adresy rozszerzone

Rozszerzony sposób pisania adresów stwarza możliwość uporządkowania w ramach systematyki dużej ilości osi i wrzecion. Adres rozszerzony składa się z rozszerzenia numerycznego albo z pisanego w nawiasach kwadratowych identyfikatora zmiennej i wyrażenia arytmetycznego przyporządkowanego przy pomocy znaku "=".

Przykład:

X7	; "=" nie jest wymagane, 7 jest wartością, znak "=" jest jednak również tutaj możliwy
X4=20	;oś X4 ("=" wymagane)
CR=7.3	;2 litery ("=" wymagane)
S1=470	;prędkość obrotowa dla 1. wrzeciona 470 obr./min
M3=5	;zatrzymanie wrzeciona dla 3. wrzeciona

Rozszerzony sposób pisania adresów jest możliwy tylko dla następujących prostych adresów:

X, Y, Z, ...	Adresy osi
I, J, K	Parametry interpolacji
S	Prędkość obrotowa wrzeciona
SPOS, SPOSA	Pozycja wrzeciona
M	Funkcje dodatkowe
H	Funkcje pomocnicze
T	Numer narzędzia
F	Posuw

Liczbę (indeks) przy rozszerzonym pisaniu adresów można w przypadku adresów M, H, S jak też SPOS i SPOSA zastąpić zmienną. Identyfikator zmiennej znajduje się przy tym w nawiasach kwadratowych.

Przykład:

S[SPINU]=470 ;Prędkość obrotowa dla wrzeciona, którego numer jest zapisany w zmiennej SPINU
M[SPINU]=3 ;Obroty w prawo wrzeciona, którego numer jest zapisany w zmiennej SPINU
T[SPINU]=7 ;Wybór wstępny narzędzia dla wrzeciona, którego numer jest zapisany w zmiennej SPINU

Adresy stałe

Następujące adresy są ustalone na stałe:

Adres	Znaczenie (nastawienie standardowe)
D	Numer ostrza
F	Posuw
G	Warunek drogowy
H	Funkcja pomocnicza
L	Wywołanie podprogramu
M	Funkcja dodatkowa
N	Blok pomocniczy
P	Liczba przebiegów programu
R	Parametr obliczeniowy
S	Prędkość obrotowa wrzeciona
T	Numer narzędzia
:	Blok główny

2.6 Przykład programowania, toczenie

Przykład programowania:

N10 G54 T9 D2

Adresy stałe z rozszerzeniem osi

Adres	Znaczenie (nastawienie standardowe)
AX	Wartość osi (zmienne programowanie osi)
ACC	Przyśpieszenie osiowe
FA	Posuw osiowy
FDA	Posuw osiowy ze zmianą prędkości kółkiem ręcznym
FL	Osiowe ograniczenie posuwu
IP	Parametr interpolacji (zmienne programowanie osi)
OVRA	Override osiowy
PO	Współczynnik wielomianowy
POS	Oś pozycjonowania
POSA	Oś pozycjonowania poza granicę bloku

Przykład: N10 POS[X]=100

W przypadku programowania z rozszerzeniem osi oś, w której ma zostać wykonany ruch, znajduje się w nawiasach kwadratowych.



Kompletną listę adresów nastawionych na stałe znajdziecie w aneksie.

Adresy nastawne

Adresy mogą być definiowane albo jako litera adresowa (ewent. z numerycznym rozszerzeniem) albo jako dowolny identyfikator.



Adresy nastawne muszą być jednoznaczne w ramach sterowania, tzn. ten sam identyfikator adresu nie może być stosowany dla różnych typów adresów.

Jako typy adresów rozróżnia się przy tym:

- wartości osi i punkty końcowe
- parametry interpolacji
- posuwu
- kryteria ścięcia narożnika
- pomiar
- ruch w osi i ruch wrzeciona
- ...

Nastawne litery adresowe to:

A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z



Nazwy adresów nastawnych mogą być zmieniane przez użytkownika poprzez dane maszynowe.

Przykład:

X1, Y30, U2, I25, E25, E1=90, ...

Rozszerzenie numeryczne ma jedno lub dwa miejsca i jest zawsze dodatnie.

Identyfikator adresowy:

Sposób pisania adresów może zostać rozszerzony przez dodanie dalszych liter.

Przykład:

CR np. dla promienia okręgu
XPOS



Operatory/funkcje obliczeniowe

+	Dodawanie
-	Odejmowanie
*	Mnożenie
/	Dzielenie
	Uwaga: (typ INT)/(typ INT)=(typ REAL); np. 3/4 = 0.75
DIV	Dzielenie, dla typu zmiennej INT i REAL
	Uwaga: (typ INT)DIV(typ INT)=(typ INT); np. 3 DIV 4 = 0
MOD	Dzielenie modulo (tylko dla typu INT) daje resztę z dzielenia INT, np. 3 MOD 4=3
:	Operator powiązania (w przypadku zmiennych FRAME)
SIN()	Sinus
COS()	Cosinus
TAN()	Tangens
ASIN()	Arcus sinus
ACOS()	Arcus cosinus
ATAN2()	Arcus tangens2
SQRT()	Pierwiastek kwadratowy
ABS()	Wartość bezwzględna
POT()	2. potęga (kwadrat)
TRUNC()	Część całkowitoliczbowa
ROUND()	Zaokrąglenie do liczby całkowitej
LN()	Logarytm naturalny
EXP()	Funkcja wykładnicza



Operatory porównania i operatory logiczne

==	równe
<>	nierówne
>	większe od
<	mniejsze od
>=	większe albo równe
<=	mniejsze albo równe
AND	I
OR	LUB
NOT	Negacja
XOR	ALBO

W wyrażeniach arytmetycznych można przy pomocy nawiasów okrągłych ustalić kolejność wykonywania wszystkich operatorów a przez to czynić odstępstwa od normalnych zasad pierwszeństwa.

Przyporządkowania wartości

Adresom mogą być przyporządkowywane wartości. Przyporządkowanie wartości następuje zależnie od rodzaju identyfikatora adresu.

Znak "=" między identyfikatorem adresu i wartością musi być pisany, gdy

- identyfikator adresu składa się z więcej niż jednej litery
- wartość składa się z więcej niż jednej stałej.

Znaku "=" można nie pisać, gdy identyfikator adresu jest jedną literą a wartość składa się tylko z jednej stałej. Znaki plus i minus są dozwolone, znak rozdzielający po literze adresowej jest dopuszczalny.

Przykłady:

X10	;Przyporządkowanie wartości (10) do adresu X, "=" nie wymagane
X1=10	;Przyporządkowanie wartości (10) do adresu (X) z rozszerzeniem numerycznym (1), "=" wymagane
FGROUP (X1, Y2)	;Nazwy osi z parametrów przekazania
AXDATA[X1]	;Nazwa osi jako indeks przy dostępie do danych osi
AX[X1]=10	;Pośrednie programowanie osi
X=10*(5+SIN(37.5))	;Przyporządkowanie wartości poprzez wyrażenie numeryczne, "=" wymagane



Po rozszerzeniu numerycznym musi zawsze następować jeden ze znaków specjalnych "=", "(", "[", ")", "]", ";", " " albo operator, aby rozróżnić identyfikator adresowy z rozszerzeniem numerycznym od litery adresowej z wartością.

Identyfikatory

Słowa (według DIN 66025) są uzupełniane przez identyfikatory (nazwy). Te uzupełnienia posiadają w ramach bloku NC takie samo znaczenie jak słowa. Identyfikatory muszą być jednoznaczne. Tych samych identyfikatorów nie wolno jest stosować do różnych obiektów.

Identyfikatory mogą oznaczać:

- zmienną
- zmienną systemową
- zmienną użytkownika
- podprogramy
- słowa kluczowe
- adresy DIN z wieloma literami
- etykiety

Budowa

Identyfikatory są tworzone z maksymalnie 32 znaków.

Jako znaki wolno stosować:

- litery
- podkreślniki
- cyfry

Dwa pierwsze znaki muszą być literami albo podkreślnikami, między poszczególnymi znakami nie może być znaków rozdzielających (patrz następne strony).

Przykład: C MIRROR, CDON



Zarezerwowanych słów kluczowych nie wolno używać jako identyfikatorów. Znaki rozdzielające między poszczególnymi znakami są niedozwolone.



Liczba znaków dla poszczególnych identyfikatorów:

- nazwa programu: 24 znaki
- identyfikator osi 8 znaków
- identyfikator zmiennej 31 znaków

Zasady nadawania nazw identyfikatorów

W celu uniknięcia kolizji nazw przyjmuje się następujące uregulowanie:

- Wszystkie identyfikatory, które rozpoczynają się od "CYCLE" albo "_", są zarezerwowane dla cykli SIEMENS.
- Wszystkie identyfikatory, które rozpoczynają się od "CCS" są zarezerwowane dla cykli kompilacyjnych SIEMENS.
- Cykle kompilacyjne użytkownika rozpoczynają się od "CC".
- Użytkownikom zalecamy wybieranie nazw identyfikatorów, które rozpoczynają się od "U" (user) albo zawierają podkreślniki, ponieważ takie identyfikatory nie są używane przez system, cykle kompilacyjne ani cykle SIEMENS.

Dalsze rezerwacje

- Identyfikator "RL" jest zarezerwowany dla tokarek konwencjonalnych.
- Identyfikatory, które rozpoczynają się od "E_", są zarezerwowane w przypadku programowania EASY-STEP.

Identyfikatory zmiennych

W przypadku zmiennych, które są używane przez system, pierwsza litera jest zastępowana przez znak "\$". Dla zmiennych definiowanych przez użytkownika nie wolno jest używać tego znaku.

Przykłady (patrz "Przygotowanie pracy"):

```
$P_IFRAME, $P_F
```

W przypadku zmiennych z rozszerzeniem numerycznym zera na początku nie mają znaczenia (R01 odpowiada R1). Przed rozszerzeniem numerycznym są dozwolone znaki rozdzielające.